# Semi-Supervised Cause Identification from Aviation Safety Reports

**Isaac Persing** and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
{persingq,vince}@hlt.utdallas.edu

## Abstract

We introduce cause identification, a new problem involving classification of incident reports in the aviation domain. Specifically, given a set of pre-defined causes, a cause identification system seeks to identify all and only those causes that can explain why the aviation incident described in a given report occurred. The difficulty of cause identification stems in part from the fact that it is a multi-class, multi-label categorization task, and in part from the skewness of the class distributions and the scarcity of annotated reports. To improve the performance of a cause identification system for the minority classes, we present a bootstrapping algorithm that automatically augments a training set by learning from a small amount of labeled data and a large amount of unlabeled data. Experimental results show that our algorithm yields a relative error reduction of 6.3% in F-measure for the minority classes in comparison to a baseline that learns solely from the labeled data.

## 1 Introduction

Automatic text classification is one of the most important applications in natural language processing (NLP). The difficulty of a text classification task depends on various factors, but typically, the task can be difficult if (1) *the amount of labeled data* available for learning the task is small; (2) it involves multiple classes; (3) it involves *multi-label categorization*, where more than one label can be assigned to each document; (4) the *class distributions* are skewed, with some categories significantly outnumbering the others; and (5) the documents belong to the *same domain* (e.g., movie review classification). In particular, when the documents to be classified are from the same domain, they tend to be more similar to each other with respect to word usage, thus making the classes less easily separable. This is one of the reasons why topic-based classification, even with multiple classes as in the 20 Newsgroups dataset[1], tends to be easier than review classification, where reviews from the same domain are to be classified according to the sentiment expressed[2].

In this paper, we introduce a new text classification problem involving the Aviation Safety Reporting System (ASRS) that can be viewed as a difficult task along each of the five dimensions discussed above. Established in 1967, ASRS collects voluntarily submitted reports about aviation safety incidents written by flight crews, attendants, controllers, and other related parties. These incident reports are made publicly available to researchers for automatic analysis, with the ultimate goal of improving the aviation safety situation. One central task in the automatic analysis of these reports is *cause identification*, or the identification of *why* an incident happened. Aviation safety experts at NASA have identified 14 causes (or *shaping factors* in NASA terminology) that could explain why an incident occurred. Hence, cause identification can be naturally recast as a text classification task: given an incident report, determine which of a set of 14 shapers contributed to the occurrence of the incident described in the report.

As mentioned above, cause identification is considered challenging along each of the five aforementioned dimensions. First, there is a scarcity of incident reports labeled with the shapers. This can be attributed to the fact that there has been very little work on this task. While the NASA researchers have applied a heuristic method for labeling a report with shapers (Posse

[1] http://kdd.ics.uci.edu/databases/20newsgroups/

[2] Of course, the fact that sentiment classification requires a deeper understanding of a text also makes it more difficult than topic-based text classification (Pang et al., 2002).

et al., 2005), the method was evaluated on only 20 manually labeled reports, which are not made publicly available. Second, the fact that this is a 14-class classification problem makes it more challenging than a binary classification problem. Third, a report can be labeled with more than one category, as several shapers can contribute to the occurrence of an aviation incident. Fourth, the class distribution is very skewed: based on an analysis of our 1,333 annotated reports, 10 of the 14 categories can be considered minority classes, which account for only 26% of the total number of labels associated with the reports. Finally, our cause identification task is domain-specific, involving the classification of documents that all belong to the aviation domain.

This paper focuses on improving the accuracy of minority class prediction for cause identification. Not surprisingly, when trained on a dataset with a skewed class distribution, most supervised machine learning algorithms will exhibit good performance on the majority classes, but relatively poor performance on the minority classes. Unfortunately, achieving good accuracies on the minority classes is very important in our task of identifying shapers from aviation safety reports, where 10 out of the 14 shapers are minority classes, as mentioned above. Minority class prediction has been tackled extensively in the machine learning literature, using methods that typically involve sampling and re-weighting of training instances, with the goal of creating a less skewed class distribution (e.g., Pazzani et al. (1994), Fawcett (1996), Kubat and Matwin (1997)). Such methods, however, are unlikely to perform equally well for our cause identification task given our small labeled set, as the minority class prediction problem is complicated by the scarcity of labeled data. More specifically, given the scarcity of labeled data, many words that are potentially correlated with a shaper (especially a minority shaper) may not appear in the training set, and the lack of such useful indicators could hamper the acquisition of an accurate classifier via supervised learning techniques.

We propose to address the problem of minority class prediction in the presence of a small training set by means of a bootstrapping approach, where we introduce an iterative algorithm to (1) use a small set of labeled reports and a large set of unlabeled reports to automatically identify words that are most relevant to the minority shaper under consideration, and (2) augment the labeled data by using the resulting words to annotate those unlabeled reports that can be confidently labeled. We evaluate our approach using cross-validation on 1,333 manually annotated reports. In comparison to a supervised baseline approach where a classifier is acquired solely based on the training set, our bootstrapping approach yields a relative error reduction of 6.3% in F-measure for the minority classes.

In sum, the contributions of our work are threefold. First, we introduce a new, challenging text classification problem, cause identification from aviation safety reports, to the NLP community. Second, we created an annotated dataset for cause identification that is made publicly available for stimulating further research on this problem[3]. Third, we introduce a bootstrapping algorithm for improving the prediction of minority classes in the presence of a small training set.

The rest of the paper is organized as follows. In Section 2, we present the 14 shapers. Section 3 explains how we preprocess and annotate the reports. Sections 4 and 5 describe the baseline approaches and our bootstrapping algorithm, respectively. We present results in Section 6, discuss related work in Section 7, and conclude in Section 8.

## 2 Shaping Factors

As mentioned in the introduction, the task of cause identification involves labeling an incident report with all the shaping factors that contributed to the occurrence of the incident. Table 1 lists the 14 shaping factors, as well as a description of each shaper taken verbatim from Posse et al. (2005). As we can see, the 14 classes are not mutually exclusive. For instance, a lack of familiarity with equipment often implies a deficit in proficiency in its use, so the two shapers frequently co-occur. In addition, while some classes cover a specific and well-defined set of issues (e.g., Illusion), some encompass a relatively large range of situations. For instance, resource deficiency can include problems with equipment, charts, or even aviation personnel. Furthermore, ten shaping factors can be considered minority classes, as each of them account for less than 10% of the labels. Accurately predicting minority classes is important in this domain because, for example, the physical factors minority shaper is frequently associated with incidents involving near-misses between aircraft.

[3] http://www.hlt.utdallas.edu/∼persingq/ASRSdataset.html

| Id | Shaping Factor | Description | % |
|---|---|---|---|
| 1 | **Attitude** | Any indication of unprofessional or antagonistic attitude by a controller or flight crew member, e.g., complacency or get-homeitis (in a hurry to get home). | 2.4 |
| 2 | **Communication Environment** | Interferences with communications in the cockpit such as noise, auditory interference, radio frequency congestion, or language barrier. | 5.5 |
| 3 | **Duty Cycle** | A strong indication of an unusual working period, e.g., a long day, flying very late at night, exceeding duty time regulations, having short and inadequate rest periods. | 1.8 |
| 4 | **Familiarity** | A lack of factual knowledge, such as new to or unfamiliar with company, airport, or aircraft. | 3.2 |
| 5 | **Illusion** | Bright lights that cause something to blend in, black hole, white out, sloping terrain, etc. | 0.1 |
| 6 | **Other** | Anything else that could be a shaper, such as shift change, passenger discomfort, or disorientation. | 13.3 |
| 7 | **Physical Environment** | Unusual physical conditions that could impair flying or make things difficult. | 16.0 |
| 8 | **Physical Factors** | Pilot ailment that could impair flying or make things more difficult, such as being tired, drugged, incapacitated, suffering from vertigo, illness, dizziness, hypoxia, nausea, loss of sight or hearing. | 2.2 |
| 9 | **Preoccupation** | A preoccupation, distraction, or division of attention that creates a deficit in performance, such as being preoccupied, busy (doing something else), or distracted. | 6.7 |
| 10 | **Pressure** | Psychological pressure, such as feeling intimidated, pressured, or being low on fuel. | 1.8 |
| 11 | **Proficiency** | A general deficit in capabilities, such as inexperience, lack of training, not qualified, or not current. | 14.4 |
| 12 | **Resource Deficiency** | Absence, insufficient number, or poor quality of a resource, such as overworked or unavailable controller, insufficient or out-of-date chart, malfunctioning or inoperative or missing equipment. | 30.0 |
| 13 | **Taskload** | Indicators of a heavy workload or many tasks at once, such as short-handed crew. | 1.9 |
| 14 | **Unexpected** | Something sudden and surprising that is not expected. | 0.6 |

Table 1: Descriptions of shaping factor classes. The "%" column shows the percent of labels the shapers account for.

## 3 Dataset

We downloaded our corpus from the ASRS website[4]. The corpus consists of 140,599 incident reports collected during the period from January 1998 to December 2007. Each report is a free text narrative that describes not only why an incident happened, but also what happened, where it happened, how the reporter felt about the incident, the reporter's opinions of other people involved in the incident, and any other comments the reporter cared to include. In other words, a lot of information in the report is irrelevant to (and thus complicates) the task of cause identification.

### 3.1 Preprocessing

Unlike newswire articles, at which many topic-based text classification tasks are targeted, the ASRS reports are informally written using various domain-specific abbreviations and acronyms, tend to contain poor grammar, and have capitalization information removed, as illustrated in the following sentence taken from one of the reports.

> HAD BEEN CLRED FOR APCH BY ZOA AND HAD BEEN HANDED OFF TO SANTA ROSA TWR.

This sentence is grammatically incorrect (due to the lack of a subject), and contains abbreviations such as CLRED, APCH, and TWR. This makes it difficult for a non-aviation expert to understand. To improve readability (and hence facilitate the annotation process), we preprocess each report as follows. First, we expand the abbreviations/acronyms with the help of an official list of acronyms/abbreviations and their expanded forms[5]. Second, though not as crucial as the first step, we heuristically restore the case of the words by relying on an English lexicon: if a word appears in the lexicon, we assume that it is not a proper name, and therefore convert it into lowercase. After preprocessing, the example sentence appears as

> had been cleared for approach by ZOA and had been handed off to santa rosa tower.

Finally, to facilitate automatic analysis, we stem each word in the narratives.

### 3.2 Human Annotation

Next, we randomly picked 1,333 preprocessed reports and had two graduate students not affiliated

[4] http://asrs.arc.nasa.gov/

[5] See http://akama.arc.nasa.gov/ASRSDBOnline/pdf/ASRS_Decode.pdf. In the very infrequently-occurring case where the same abbreviation or acronym may have more than expansion, we arbitrarily chose one of the possibilities.

| Id | Total (%) | F1 | F2 | F3 | F4 | F5 |
|---|---|---|---|---|---|---|
| 1 | 52 (3.9) | 11 | 7 | 7 | 17 | 10 |
| 2 | 119 (8.9) | 29 | 29 | 22 | 16 | 23 |
| 3 | 38 (2.9) | 10 | 5 | 6 | 9 | 8 |
| 4 | 70 (5.3) | 11 | 12 | 9 | 14 | 24 |
| 5 | 3 (0.2) | 0 | 0 | 0 | 1 | 2 |
| 6 | 289 (21.7) | 76 | 44 | 60 | 42 | 67 |
| 7 | 348 (26.1) | 73 | 63 | 82 | 59 | 71 |
| 8 | 48 (3.6) | 11 | 14 | 8 | 11 | 4 |
| 9 | 145 (10.9) | 29 | 25 | 38 | 28 | 25 |
| 10 | 38 (2.9) | 12 | 10 | 4 | 7 | 5 |
| 11 | 313 (23.5) | 65 | 50 | 74 | 46 | 78 |
| 12 | 652 (48.9) | 149 | 144 | 125 | 123 | 111 |
| 13 | 42 (3.2) | 7 | 8 | 8 | 6 | 13 |
| 14 | 14 (1.1) | 3 | 3 | 3 | 3 | 2 |

Table 2: Number of occurrences of each shaping factor in the dataset. The "Total" column shows the number of narratives labeled with each shaper and the percentage of narratives tagged with each shaper in the 1,333 labeled narrative set. The "F" columns show the number narratives associated with each shaper in folds F1 - F5.

| $x$ **(# Shapers)** | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Percentage** | 53.6 | 33.2 | 10.3 | 2.7 | 0.2 | 0.1 |

Table 3: Percentage of documents with $x$ labels.

with this research independently annotate them with shaping factors, based solely on the definition of each shaper presented in Table 1. To measure inter-annotator agreement, we compute Cohen's Kappa (Carletta, 1996) from the two sets of annotations, obtaining a Kappa value of only 0.43. This not only suggests the difficulty of the cause identification task, but also reveals the vagueness inherent in the definition of the 14 shapers. As a result, we had the two annotators re-examine each report for which there was a disagreement and reach an agreement on its final set of labels. Statistics of the annotated dataset can be found in Table 2, where the "Total" column shows the size of each of the 14 classes, expressed both as the number of reports that are labeled with a particular shaper and as a percent (in parenthesis). Since we will perform 5-fold cross validation in our experiments, we also show the number of reports labeled with each shaper under the "F" columns for each fold. To get a better idea of how many reports have multiple labels, we categorize the reports according to the number of labels they contain in Table 3.

## 4 Baseline Approaches

In this section, we describe two baseline approaches to cause identification. Since our ultimate goal is to evaluate the effectiveness of our bootstrapping algorithm, the baseline approaches only make use of small amounts of labeled data for acquiring classifiers. More specifically, both baselines recast the cause identification problem as a set of 14 binary classification problems, one for predicting each shaper. In the binary classification problem for predicting shaper $s_i$, we create one training instance from each document in the training set, labeling the instance as positive if the document has $s_i$ as one of its labels, and negative otherwise. After creating training instances, we train a binary classifier, $c_i$, for predicting $s_i$, employing as features the top 50 unigrams that are selected according to information gain computed over the training data (see Yang and Pedersen (1997) for details). The SVM learning algorithm as implemented in the LIBSVM software package (Chang and Lin, 2001) is used for classifier training, owing to its robust performance on many text classification tasks.

In our first baseline, we set all the learning parameters to their default values. As noted before, we divide the 1,333 annotated reports into five folds of roughly equal size, training the classifiers on four folds and applying them separately to the remaining fold. Results are reported in terms of precision (P), recall (R), and F-measure (F), which are computed by aggregating over the 14 shapers as follows. Let $tp_i$ be the number of test reports correctly labeled as positive by $c_i$; $p_i$ be the total number of test reports labeled as positive by $c_i$; and $n_i$ be the total number of test reports that belong to $s_i$ according to the gold standard. Then,

$$\mathrm{P} = \frac{\sum_i tp_i}{\sum_i p_i}, \mathrm{R} = \frac{\sum_i tp_i}{\sum_i n_i}, \text{and } \mathrm{F} = \frac{2PR}{P+R}.$$

Our second baseline is similar to the first, except that we tune the classification threshold (CT) to optimize F-measure. More specifically, recall that LIBSVM trains a classifier that by default employs a CT of 0.5, thus classifying an instance as positive if and only if the probability that it belongs to the positive class is at least 0.5. However, this may not be the optimal threshold to use as far as performance is concerned, especially for the minority classes, where the class distribution is skewed. This is the motivation behind tuning the CT of each classifier. To ensure a fair comparison with the first baseline, we do not employ additional labeled data for parameter tuning; rather, we reserve 25% of the available training data for

tuning, and use the remaining 75% for classifier acquisition. This amounts to using three folds for training and one fold for development in each cross validation experiment. Using the development data, we tune the 14 CTs *jointly* to optimize overall F-measure. However, an exact solution to this optimization problem is computationally expensive. Consequently, we find a local maximum by employing a local search algorithm, which alters one parameter at a time to optimize F-measure by holding the remaining parameters fixed.

## 5 Our Bootstrapping Algorithm

One of the potential weaknesses of the two baselines described in the previous section is that the classifiers are trained on only a small amount of labeled data. This could have an adverse effect on the accuracy of the resulting classifiers, especially those for the minority classes. The situation is somewhat aggravated by the fact that we are adopting a one-versus-all scheme for generating training instances for a particular shaper, which, together with the small amount of labeled data, implies that only a couple of positive instances may be available for training the classifier for a minority class. To alleviate the data scarcity problem and improve the accuracy of the classifiers, we propose in this section a bootstrapping algorithm that automatically augments a training set by exploiting a large amount of unlabeled data. The basic idea behind the algorithm is to iteratively identify words that are high-quality indicators of the positive or negative examples, and then automatically label unlabeled documents that contain a sufficient number of such indicators.

Our bootstrapping algorithm, shown in Figure 1, aims to augment the set of positive and negative training instances for a given shaper. The main function, *Train*, takes as input four arguments. The first two arguments, $P$ and $N$, are the positive and negative instances, respectively, generated by the one-versus-one scheme from the initial training set, as described in the previous section. The third argument, $U$, is the unlabeled set of documents, which consists of all but the documents in the training set. In particular, $U$ contains the documents in the development and test sets. Hence, we are essentially assuming access to the test documents (but not their labels) during the training process, as in a transductive learning setting. The last argument, $k$, is the number of bootstrapping iterations. In addition, the algorithm uses two variables, $PW$ and $NW$, to store the sets of high-quality indicators for the positive instances and the negative instances, respectively, that are found during the bootstrapping process.

$Train(P, N, U, k)$
**Inputs:**
$P$: positively labeled training examples of shaper $x$
$N$: negatively labeled training examples of shaper $x$
$U$: set of unlabeled narratives in corpus
$k$: number of bootstrapping iterations
$PW \leftarrow \emptyset$
$NW \leftarrow \emptyset$
**for** $i$ = 0 to $k-1$ **do**
  **if** $|P| > |N|$ **then**
    $[P, PW] \leftarrow ExpandTrainingSet(P, N, U, PW)$
  **else**
    $[N, NW] \leftarrow ExpandTrainingSet(N, P, U, NW)$
  **end if**
**end for**

---

$ExpandTrainingSet(A, B, U, W)$
**Inputs:**
$A, B, U$: narrative sets
$W$: unigram feature set
**for** $j$ = 1 to 4 **do**
  $t \leftarrow \arg\max_{t \notin W} \left(\log(\frac{C(t,A)}{C(t,B)+1})\right)$
  // $C(t, X)$: number of narratives in $X$ containing $t$
  $W \leftarrow W \cup \{t\}$
**end for**
return $[A \cup S(W, U), W]$
// $S(W, U)$: narratives in $U$ containing $\geq 3$ words in $W$

Figure 1: Our bootstrapping algorithm.

Next, we begin our $k$ bootstrapping iterations. In each iteration, we expand either $P$ or $N$, depending on their relative sizes. In order to keep the two sets as close in size as possible, we choose to expand the smaller of the two sets. After that, we execute the function *ExpandTrainingSet* to expand the selected set. Without loss of generality, assume that $P$ is chosen for expansion. To do this, *ExpandTrainingSet* selects four words that seem much more likely to appear in $P$ than in $N$ from the set of candidate words[6]. To select these words, we calculate the log likelihood ratio $\log(\frac{C(t,P)}{C(t,N)+1})$ for each candidate word $t$, where $C(t, P)$ is the number of narratives in $P$ that contain $t$, and $C(t, N)$ similarly is the number of narratives in $N$ that contain $t$. If this ratio is large, we posit that $t$ is a good indicator of $P$. Note that incrementing the count in the denominator by one

[6] A candidate word is a word that appears in the training set ($P \cup N$) at least four times.

has a smoothing effect: it avoids selecting words that appears infrequently in $P$ and not at all in $N$.

There is a reason for selecting multiple words (rather than just one word) in each bootstrapping iteration: we want to prevent the algorithm from selecting words that are too specific to one subcategory of a shaping factor. For example, shaping factor 7 (Physical Environment) is composed largely of incidents influenced by weather phenomena. In one experiment, we tried selecting only one word per bootstrapping iteration. For shaper 7, the first word added to PW was "snow". Upon the next iteration, the algorithm added "plow" to PW. While "plow" may itself be indicative of shaper 7, we believe its selection was due to the recent addition to P of a large number of narratives containing "snow". Hence, by selecting four words per iteration, we are forcing the algorithm to "branch out" among these subcategories.

After adding the selected words to $PW$, we augment $P$ with all the unlabeled documents containing at least three words from $PW$. The reason for imposing the "at least three" requirement is precision: we want to ensure, with a reasonable level of confidence, that the unlabeled documents chosen for augmenting $P$ should indeed be labeled with the shaper under consideration, as incorrectly labeled documents would contaminate the labeled data, thus accelerating the deterioration of the quality of the automatically labeled data in subsequent bootstrapping iterations and adversely affecting the accuracy of the classifier trained on it (Pierce and Cardie, 2001).

The above procedure is repeated in each bootstrapping iteration. As mentioned above, if $N$ is smaller in size than $P$, we will expand $N$ instead, adding to $NW$ the four words that are the strongest indicators of a narrative being a negative example of the shaper under consideration, and augmenting $N$ with those unlabeled narratives that contain at least three words from $NW$.

For a typical minority shaper, the algorithm expands P the first two iterations. On the second iteration, P is augmented with the narratives in the unlabeled set containing any 3 of the 8 words in PW. Given the unlabeled set's ample size, these documents can easily number in the hundreds, tipping the balance between $|P|$ and $|N|$ so that in the next iteration, $|N| > |P|$.

The number of bootstrapping iterations is controlled by the input parameter $k$. As we will see in the next section, we run the bootstrapping algorithm for up to five iterations only, as the quality of the bootstrapped data deteriorates fairly rapidly. The exact value of $k$ will be determined automatically using development data, as discussed below.

After bootstrapping, the augmented training data can be used in combination with any of the two baseline approaches to acquire a classifier for identifying a particular shaper. Whichever baseline is used, we need to reserve one of the five folds to tune the parameter $k$ in our cross validation experiments. In particular, if the second baseline is used, we will tune $CT$ and $k$ jointly on the development data using the local search algorithm described previously, where we adjust the values of both $CT$ and $k$ for one of the 14 classifiers in each step of the search process to optimize the overall F-measure score.

## 6 Evaluation

### 6.1 Baseline Systems

Since our evaluation centers on the question of how effective our bootstrapping algorithm is in exploiting unlabeled documents to improve classifier performance, our two baselines only employ the available labeled documents to train the classifiers.

Recall that our first baseline, which we call $B_{0.5}$ (due to its being a baseline with a CT of 0.5), employs default values for all of the learning parameters. Micro-averaged 5-fold cross validation results of this baseline for all 14 shapers and for just 10 minority classes (due to our focus on improving minority class prediction) are expressed as percentages in terms of precision (P), recall (R), and F-measure (F) in the first row of Table 4. As we can see, the baseline achieves an F-measure of 45.4 (14 shapers) and 35.4 (10 shapers). Comparing these two results, the higher F-measure achieved using all 14 shapers can be attributed primarily to improvements in recall. This should not be surprising: as mentioned above, the number of positive instances created for a minority class could be small, thus causing the resulting classifier to be biased towards classifying a document as negative.

Instead of employing a CT value of 0.5, our second baseline, $B_{ct}$, tunes the CT using one of the training folds and simply trains a classifier on the remaining three folds. For parameter tuning, we tested CTs of 0.0, 0.05, ..., 1.0. Results of this baseline are shown in row 2 of Table 4. In com-

| System | All 14 Classes | | | 10 Minority Classes | | |
|---|---|---|---|---|---|---|
| | P | R | F | P | R | F |
| $\mathbf{B}_{0.5}$ | 67.0 | 34.4 | 45.4 | 68.3 | 23.9 | 35.4 |
| $\mathbf{B}_{ct}$ | 47.4 | 59.2 | 52.7 | 47.8 | 34.3 | 39.9 |
| $\mathbf{E}_{0.5}$ | 60.9 | 40.4 | 48.6 | 53.2 | 35.3 | 42.4 |
| $\mathbf{E}_{ct}$ | 50.5 | 54.9 | 52.6 | 49.1 | 39.4 | 43.7 |

Table 4: 5-fold cross validation results.

parison to the first baseline, we see that F-measure improves considerably by 7.4% and 4.5% for 14 shapers and 10 shapers[7], respectively, which illustrates the importance of employing the right CT for the cause identification task.

### 6.2 Our Approach

Next, we evaluate the effectiveness of our bootstrapping algorithm in improving classifier performance. More specifically, we apply the two baselines separately to the augmented training set produced by our bootstrapping algorithm. When combining our bootstrapping algorithm with the first baseline, we produce a system that we call $E_{0.5}$ (due to its being trained on the *expanded* training set with a CT of 0.5). $E_{0.5}$ has only one tunable parameter, $k$ (i.e., the number of bootstrapping iterations), whose allowable values are 0, 1, . . ., 5. When our algorithm is used in combination with the second baseline, we produce another system, $E_{ct}$, which has both $k$ and the CT as its parameters. The allowable values of these parameters, which are to be tuned jointly, are the same as those employed by $B_{ct}$ and $E_{0.5}$.

Results of $E_{0.5}$ are shown in row 3 of Table 4. In comparison to $B_{0.5}$, we see that F-measure increases by 3.2% and 7.0% for 14 shapers and 10 shapers, respectively. Such increases can be attributed to less imbalanced recall and precision values, as a result of a large gain in recall accompanied by a roughly equal drop in precision. These results are consistent with our intuition: recall can be improved with a larger training set, but precision can be hampered when learning from noisily labeled data. Overall, these results suggest that learning from the augmented training set is useful, especially for the minority classes.

Results of $E_{ct}$ are shown in row 4 of Table 4. In comparison to $B_{ct}$, we see mixed results: F-measure increases by 3.8% for 10 shapers (which represents a relative error reduction of 6.3%, but drops by 0.1% for 14 shapers. Overall, these results suggest that when the CT is tunable, training set expansion helps the minority classes but hurts the remaining classes. A closer look at the results reveals that the 0.1% F-measure drop is due to a large drop in recall accompanied by a smaller gain in precision. In other words, for the four non-minority classes, the benefits obtained from using the bootstrapped documents can also be obtained by simply adjusting the CT. This could be attributed to the fact that a decent classifier can be trained using only the hand-labeled training examples for these four shapers, and as a result, the automatically labeled examples either provide very little new knowledge or are too noisy to be useful. On the other hand, for the 10 minority classes, the 3.8% gain in F-measure can be attributed to a simultaneous rise in recall and precision. Note that such gain cannot possibly be obtained by simply adjusting the CT, since adjusting the CT always results in higher recall and lower precision or vice versa. Overall, the simultaneous rise in recall and precision implies that the bootstrapped documents have provided useful knowledge, particularly in the form of positive examples, for the classifiers. Even though the bootstrapped documents are noisily labeled, they can still be used to improve the classifiers, as the set of initially labeled positive examples for the minority classes is too small.

### 6.3 Additional Analyses

**Quality of the bootstrapped data.** Since the bootstrapped documents are noisily labeled, a natural question is: how noisy are they? To answer this question, we need to label all of the bootstrapped documents. To get a sense of the accuracy of the bootstrapped documents without further manual labeling, recall that our experimental setup resembles a transductive setting where the test documents are part of the unlabeled data, and consequently, some of them may have been automatically labeled by the bootstrapping algorithm. In fact, 137 documents in the five test folds were automatically labeled in the 14-shaper $E_{ct}$ experiments, and 69 automatically labeled documents were similarity obtained from the 10-shaper $E_{ct}$ experiments. For 14 shapers, the accuracies of the positively and negatively labeled documents are 74.6% and 97.1%, respectively, and the corresponding numbers for 10 shapers are 43.2% and

[7] It is important to note that the parameters are optimized separately for each pair of 14-shaper and 10-shaper experiments in this paper, and that the 10-shaper results are not simply extracted from the 14-shaper experiments.

| Shaping Factor | Positive Expanders | Negative Expanders |
|---|---|---|
| **Familiarity** | unfamiliar, layout, unfamilarity, rely | |
| **Physical Environment** | cloud, snow, ice, wind | |
| **Physical Factors** | fatigue, tire, night, rest, hotel, awake, sleep, sick | declare, emergency, advisory, separation |
| **Preoccupation** | distract, preoccupied, awareness, situational, task, interrupt, focus, eye, configure, sleep | declare, ice snow, crash, fire, rescue, anti, smoke |
| **Pressure** | bad, decision, extend, fuel, calculate, reserve, diversion, alternate | |

Table 5: Example positive and negative expansion words collected by $E_{ct}$ for selected shaping factors.

81.3%. These numbers suggest that negative examples can be acquired with high accuracies, but the same is not true for positive examples. Nevertheless, learning the 10 shapers from the not-so-accurately-labeled positive examples still allows us to outperform the corresponding baseline.

**Analysis of the expanders.** To get an idea of whether the words acquired during the bootstrapping process (henceforth *expanders*) make intuitive sense, we show in Table 5 example positive and negative expanders obtained for five shaping factors from the $E_{ct}$ experiments. As we can see, many of the positive expanders are intuitively obvious. We might, however, wonder about the connection between, for example, the shaper Familiarity and the word "rely", or between the shaper Pressure and the word "extend". We suspect that the bootstrapping algorithm is likely to make poor word selections particularly in the cases of the minority classes, where the positively labeled training data used to select expansion words is more sparse. As suggested earlier, poor word choice early in the algorithm is likely to cause even poorer word choice later on.

On the other hand, while none of the negative expanders seem directly meaningful in relation to the shaper for which they were selected, some of them do appear to be related to other phenomena that may be negatively correlated with the shaper. For instance, the words "snow" and "ice" were selected as negative expanders for Preoccupation and also as positive expanders for Physical Environment. While these two shapers are only slightly negatively correlated, it is possible that Preoccupation may be strongly negatively correlated with the subset of Physical Environment incidents involving cold weather.

## 7 Related Work

Since we (1) recast cause identification as a text classification task and (2) proposed a bootstrapping approach that targets at improving minority class prediction, the work most related to ours involves one or more of these topics.

Guzmán-Cabrera et al. (2007) address the problem of class skewness in text classification. Specifically, they first under-sample the majority classes, and then bootstrap the classifier trained on the under-sampled data using unlabeled documents collected from the Web.

Nigam et al. (2000) propose an iterative semi-supervised method that employs the EM algorithm in combination with the naive Bayes generative model to combine a small set of labeled documents and a large set of unlabeled documents. McCallum and Nigam (1999) suggest that the initial labeled examples can be obtained using a list of keywords rather than through annotated data, yielding an unsupervised algorithm.

Similar bootstrapping methods are applicable outside text classification as well. One of the most notable examples is Yarowsky's (1995) bootstrapping algorithm for word sense disambiguation. Beginning with a list of unlabeled contexts surrounding a word to be disambiguated and a list of seed words for each possible sense, the algorithm iteratively uses the seeds to label a training set from the unlabeled contexts, and then uses the training set to identify more seed words.

## 8 Conclusions

We have introduced a new problem, cause identification from aviation safety reports, to the NLP community. We recast it as a multi-class, multi-label text classification task, and presented a bootstrapping algorithm for improving the prediction of minority classes in the presence of a small training set. Experimental results show that our algorithm yields a relative error reduction of 6.3% in F-measure over a purely supervised baseline when applied to the minority classes. By making our annotated dataset publicly available, we hope to stimulate research in this challenging problem.

## Acknowledgments

We thank the three anonymous reviewers for their invaluable comments on an earlier draft of the paper. We are indebted to Muhammad Arshad Ul Abedin, who provided us with a preprocessed version of the ASRS corpus and, together with Marzia Murshed, annotated the 1,333 documents. This work was supported in part by NASA Grant NNX08AC35A and NSF Grant IIS-0812261.

## References

Jean Carletta. 1996. Assessing agreement on classification tasks: The Kappa statistic. *Computational Linguistics*, 22(2):249–254.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: A library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Tom Fawcett. 1996. *Learning with skewed class distributions — summary of responses*. Machine Learning List: Vol. 8, No. 20.

Rafael Guzmán-Cabrera, Manuel Montes-y-Gómez, Paolo Rosso, and Luis Villaseñor Pineda. 2007. Taking advantage of the Web for text classification with imbalanced classes. In *Proceedings of MICAI*, pages 831–838.

Miroslav Kubat and Stan Matwin. 1997. Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the 14th International Conference on Machine Learning* (ICML-97), pages 179–186, Vanderbilt University, Memphis, TN. Morgan Kaufmann.

Andrew McCallum and Kamal Nigam. 1999. Text classification by bootstrapping with keywords, EM and shrinkage. In *ACL Workshop for Unsupervised Learning in Natural Language Processing*, pages 52–58.

Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

Michael Pazzani, Christopher Merz, Patrick Murphy, Kamal Ali, Timothy Hume, and Clifford Brunk. 1994. Reducing misclassification costs. In *Proceedings of ICML*, pages 217–225.

David Pierce and Claire Cardie. 2001. Limitations of co-training for natural language learning from large datasets. In *Proceedings of EMNLP*, pages 1–9.

Christian Posse, Brett Matzke, Catherine Anderson, Alan Brothers, Melissa Matzke, and Thomas Ferryman. 2005. Extracting information from narratives: An application to aviation safety reports. In *Proceedings of the Aerospace Conference 2005*, pages 3678–3690.

Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proceedings of ICML*, pages 412–420.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the ACL*, pages 189–196.